Remarks and Arguments

Claims 1-43 have been submitted for examination. Claims 1, 2, 5, 8-13, 15, 16, 18, 29, 33 and 43 have been amended.

The drawings were objected to because several reference numerals in Figure 4C were not mentioned in the specification. In response, the specification has been amended at page 9, line 5, to insert a description of the referenced elements in Figure 4C. An additional change has been made to specification at page 11, line 3, to correct a typographical error.

Claims 1-43 has been objected to for typographical errors. In response, claims 1, 15, 29 and 43 have been amended as suggested by the examiner to correct the error. Claim 30 has also been amended to correct a typographical error.

Claims 2 and 3 have been rejected under 35 U.S.C. §112, second paragraph for lack of antecedent basis for the term "code check" in line 6 (second occurrence.) Claim 3 has been rejected for depending on claim 3 and inheriting the deficiencies thereof. In response, claim 2 has been amended in line 3 to recite "generating a code check by subtracting..." The term "code check" in line 6 (second occurrence) thus finds antecedent basis in line 3. Claim 2 is therefore proper under 35 U.S.C. §112, second paragraph. Claim 3 is likewise proper.

Claim 5 has been rejected under 35 .U.S.C. §112, second paragraph, for reciting the phrase "rotating the code check" which the examiner finds indefinite since the code check consists of a set of bits. In addition, the examiner finds that there is no structural relationship between the term "rotating" and the term "code check." In response, claim 5 has been amended, in line 2, to recite "rotating bits in the code check..." Bit rotation is well-known and many processors have built-in microcode for performing such rotation. Consequently, it is believed that amended claim 5 particularly points out and distinctly claims the invention as required by 35 U.S.C. §112, second paragraph.

Claim 8 has been rejected under 35 .U.S.C. §112, second paragraph, for a lack of a structural relationship between the phrase "generating a code check..." set forth in step (a) of claim 1 (from which claim 8 depends) and the phrase "generating a one's-complement sum..." recited in claim 8. In response claim 8 has been amended to

recite, in line 2, "generating the code check as a one's-complement sum..." It is believed that the structural relationship between the recited phrases is now clear.

Claim 9 has been rejected under 35 .U.S.C. §112, second paragraph, for a lack of a structural relationship between the phrase "generating a code check..." set forth in step (b) of claim 1 (from which claim 9 depends) and the phrase "generating a one's-complement sum..." recited in claim 9. In response, claim 9 has been amended to recite, in line 2, "generating the code check as a one's-complement sum..." It is believed that the structural relationship between the recited phrases is now clear.

Claim 10 has been rejected under 35 .U.S.C. §112, second paragraph, for a lack of a structural relationship between the phrase "generating a code check..." set forth in step (a) of claim 1 (from which claim 10 depends) and the phrase "generating a term-by-term modulo-two sum..." recited in claim 10. In response, claim 10 has been amended to recite, in line 2, "generating the code check as a term-by-term modulo-two sum..." It is believed that the structural relationship between the recited phrases is now clear.

Claim 11 has been rejected under 35 .U.S.C. §112, second paragraph, for a lack of a structural relationship between the phrase "generating a code check..." set forth in step (b) of claim 1 (from which claim 11 depends) and the phrase "generating a term-by-term modulo-two sum..." recited in claim 11. In response claim 11 has been amended to recite, in line 2, "generating the code check as a term-by-term modulo-two sum..." It is believed that the structural relationship between the recited phrases is now clear.

Claim 12 has been rejected under 35 .U.S.C. §112, second paragraph, for a lack of a structural relationship between the phrase "generating a code check..." set forth in step (a) of claim 1 (from which claim 12 depends) and the phrase "generating the residue of the ingress data block..." recited in claim 12. In response claim 12 has been amended to recite, in line 2, "generating the code check as the residue of the ingress data block..." It is believed that the structural relationship between the recited phrases is now clear.

Claim 13 has been rejected under 35 .U.S.C. §112, second paragraph, for a lack of a structural relationship between the phrase "generating a code check..." set forth in step (b) of claim 1 (from which claim 13 depends) and the phrase "generating the residue of the ingress data block..." recited in claim 13. In response claim 13 has been

amended to recite, in line 2, "generating the code check as the residue of the ingress data block..." It is believed that the structural relationship between the recited phrases is now clear.

Claims 16-19 has been rejected under 35 .U.S.C. §112, second paragraph, for a lack of a structural relationship between the phrase "egress code check" recited in line 3 of claim 16 and the phrase "internal contents" recited in claim 16, line 4. In response claim 16 has been amended to recite, in lines 3 and 4, "generates an egress code check from the egress header and from intermediate contents of the outgoing encoder ..." It is believed that the structural relationship between the recited phrases is now clear, that the "intermediate contents" are digital data and that the "intermediate contents" are the contents of the outgoing encoder. See the instant specification, page 11, lines 8-17. Claims 17-19 were rejected as dependent from rejected claim 16. Accordingly, these claims are also now in proper form.

Claims 18 and 33 has been rejected under 35 .U.S.C. §112, second paragraph, for reciting the phrase "rotating the code check" which the examiner finds indefinite since the code check consists of a set of bits. In addition, the examiner finds that there is no structural relationship between the term "rotating" and the term "code check." In response, claims 18 and 33 have been amended, in lines 2 and 3, respectively, to recite "rotating bits in the code check..." Consequently, it is believed that amended claims 18 and 33 particularly point out and distinctly claim the invention as required by 35 U.S.C. §112, second paragraph.

Claims 1, 4-15, 20-29 and 32-43 were rejected under 35 U.S.C. §103(a) as obvious over U.S. Patent No. 6,157,642A (Sturza.) The examiner comments that, with regard to claim 1, step (a) is performed by the Inner Decoder 87 of Sturza; step (b) is performed by the Inner Decoder 99 and step (c) is performed by the Modulator 101. The examiner admits that Sturza does not explicitly teach the use of systematic codes. However, the examiner claims that the systematic codes can be derived from the Reed-Solomon or BCH codes that are disclosed by Sturza. The examiner concludes that it would be have obvious to use systematic codes to simplify the data recovery.

The present invention relates to a system in which data arranged in blocks is reformatted so that each original data block is broken into a plurality of new data blocks.

The purpose of the present invention is to protect the reformatted data without having to decode the original data to perform an integrity check and then to re-encode the re-formatted data. However, since the data is broken up, the new data blocks must have sizes different from the size of the original data blocks. Further, since each data block has an associated header, the header for each outgoing data block would generally be different from the original header because the original and re-formatted data block sizes are different.

In accordance with the principles of the invention a code check is calculated from, and combined with, each new data block, but the new data blocks and new code checks are both reconstituted versions of the original data blocks and the original code checks. Consequently, the data is never left without protection.

In particular, an ingress encoder computes an ingress code check from an original data block and its associated header. Then an egress encoder computes an egress code check using the same code check algorithm as the ingress encoder. The egress code check is calculated from the egress header for an outgoing data block (that is a portion of the original data block) and the data portion of the ingress code check. The outgoing information is then assembled from the egress header, the outgoing data block and the newly computed egress code check.

Thus, the invention is addressed to a completely different problem than the Sturza system. In Sturza, the data is not reformatted, and, thus, the input data block and the output data blocks are the same size and comprise the same data. In fact, Sturza performs the exact processing that the present invention is designed to avoid. At each satellite link or "hop" site depicted in Figures 6 and 7, the data is first decoded in order to perform an integrity check and/or correct errors and then re-encoded prior to transmission. Consequently, one skilled in the art would not look to the Sturza reference to solve the problem solved by the invention.

Further, the Sturza system does not correspond to the recited elements. For example, the examiner asserts that the Inner Decoder 87 of Sturza performs the generation of the code check as recited in (a) of claim 1. However, decoding involves using the code check words to check and correct errors in the data. Claim 1 (a) involves generation of the code check from the data and, thus, one skilled in the art

would see this as an encoding step rather than a decoding step. The examiner apparently claims that the Inner Decoder 87 "generates" a code check because the outer encoded payload will include a code check, mainly because the code check was included in the "data" before the inner decoding was performed. However, as used in the specification, the phrase "generating a code check" clearly involves calculating a codeword from data and appending that codeword to the data. See, for example, the instant specification, page 6, lines 1-6. In Sturza, the Inner Decoder 87 does not calculate a codeword from the "data". Rather, it removes the codeword previously calculated from the "data" and uses the codeword to integrity check and/or error correct the "data." Thus, one skilled in the art would not reasonably interpret the Inner Decoder 87 as "generating a code check" as recited in claim 1.

Further (b) and (c) of claim 1 recite that an additional code check is generated from the previously generated code check and the an egress header and then the outgoing information is formed by combining the egress header the egress data and the code check generated in step (b). The examiner claims that these steps are performed by the Inner Encoder 99 and the Modulator 101, respectively. However, claim 1 (c) states that outgoing information is formed by combining the egress header, egress data and code check. In Sturza, the data block and header are combined by the header and payload interleaver before the inner encoding takes place rather than after as recited in claim 1.

Consequently, since Sturza is directed to a different problem and has a different structure from that claimed, it cannot teach or suggest the claimed invention. Therefore, claim 1 patentably distinguishes over the cited reference.

Claims 4-14 depend, either directly or indirectly, on claim 1 and incorporate the limitations thereof. Therefore, they distinguish over the cited reference in same manner as claim 1. In addition, these claims recite additional limitations not taught or suggested by Sturza. For example, claim 4 recites that the code check generated from the ingress data is modified to compensate for non-data bits added to the ingress data block. The examiner states that Sturza discloses interleaving and that is equivalent to the recited modification. However, interleaving is a well-known process in which the order of data words in a data block is systematically changed in order to distribute burst errors.

Interleaving has nothing to do with compensating for added non-data bits. Consequently, claim 4 patentably distinguishes over the cited reference.

Claim 5 recites that the bits of the code check generated from the ingress data are rotated to compensate for non-data bits added to the ingress data block. The examiner states that a rotated Reed-Solomon code check is still a code check since the Reed-Solomon code is cyclic. However, The Sturza patent does not disclose or suggest the Reed-Solomon codes that it generates should be rotated, nor is any mechanism disclosed for performing this rotation. Consequently, claim 5 patentably distinguishes over the cited reference.

Claim 6 that the incoming data is modified to compensate for non-data bits added to the ingress data block. The examiner again equates the Sturza interleaving to this modification. As discussed with regard to claim 4, interleaving has nothing to do with compensating for added non-data bits. Consequently, claim 6 patentably distinguishes over the cited reference.

Claim 15 recites limitations parallel to those recited in claim 1. The examiner claims that the Sturza Inner Decoder 87 corresponds to the recited ingress encoder. As stated above the Inner Decoder 87 performs a decoding step not an encoding step as performed by the ingress encoder and recited in claim 1. The examiner claims that re-encoding is a process that must be performed by decoders. However, this assertion is now where taught or suggested by Sturza. The examiner is requested to provide support for this assertion if he intends to maintain it.

The examiner further asserts that the egress encoder recited in claim 15 corresponds to the Header Outer Encoder 85 of Sturza. However, the recited egress encoder generates a code check from the egress header and the ingress code check. The Sturza Header Outer Encoder generates a code check from the header information only, because the code check information has already been removed by the Header Outer Decoder 91. Thus, the Header Outer Encoder of Sturza does not correspond to the recited element.

Consequently, since Sturza is not addressed to the problem solved by the invention and the structure of Sturza does not correspond to the recited structure, Sturza cannot teach or suggest the invention claimed in Claim 15.

Claims 20-28 depend, either directly or indirectly, on claim 15 and incorporate the limitations thereof. Therefore, they distinguish over the cited reference in same manner as claim 15. In addition, these claims recite additional limitations not taught or suggested by Sturza. For example, claim 20 recites that the code check generated from the ingress data is modified to compensate for non-data bits added to the ingress data block. The examiner states that Sturza discloses interleaving and that is equivalent to the recited modification. However, as discussed above with respect to claim 4, interleaving has nothing to do with compensating for added non-data bits. Consequently, claim 20 patentably distinguishes over the cited reference.

Claim 29 contains limitations that parallel those in claim 1. As discussed above, claim 1 patentably distinguishes over the cited reference. Consequently, claim 29 distinguishes over the cited reference in the same manner as claim 1.

Claims 32-42 depend, either directly or indirectly, on claim 29 and incorporate the limitations thereof. Therefore, they distinguish over the cited reference in same manner as claim 29. In addition, these claims recite additional limitations not taught or suggested by Sturza. For example, claim 32 recites that the code check generated from the ingress data is modified to compensate for non-data bits added to the ingress data block. The examiner states that Sturza discloses interleaving and that is equivalent to the recited modification. However, as discussed above with respect to claim 4, interleaving has nothing to do with compensating for added non-data bits. Consequently, claim 32 patentably distinguishes over the cited reference.

Claim 33 recites that the bits of the code check generated from the ingress data are rotated to compensate for non-data bits added to the ingress data block. The examiner states that a rotated Reed-Solomon code check is still a code check since the Reed-Solomon code is cyclic. However, The Sturza patent does not disclose or suggest the Reed-Solomon codes that it generates should be rotated, nor is any mechanism disclosed for performing this rotation. Consequently, claim 33 patentably distinguishes over the cited reference.

Claim 34 that the incoming data is modified to compensate for non-data bits added to the ingress data block. The examiner again equates the Sturza interleaving to this modification. As discussed with regard to claim 4, interleaving has nothing to do

with compensating for added non-data bits. Consequently, claim 34 patentably distinguishes over the cited reference.

Claim 43 contains limitations that parallel those in claim 1. As discussed above, claim 1 patentably distinguishes over the cited reference. Consequently, claim 43 distinguishes over the cited reference in the same manner as claim 1.

Based on the above discussion, claims 1-43 are allowable and advancement of this application to issue is respectfully requested. The Commissioner is hereby authorized to charge any fees or credits under 37 C.F.R. §1.16 and 1.17 to our deposit account No. 02-3038.

Respectfully submitted

Paul E. Kudirka Date: 8/6/04

Paul E. Kudirka, Esq. Reg. No. 26,931
KUDIRKA & JOBSE, LLP
Customer Number 021127
Tel: (617) 367-4600 Fax: (617) 367-4656